PROGRAMMABLE FRONT-END DATA ACQUISITION MODULE FOR RADIO-XENON MONITORS

Michael Momayezi[1], Brad Hurst[1], Chien P Wong[1], Tom R. Heimbigner[2], Matthew W. Cooper[2], Charles W. Hubbard[2], and Justin I. McIntyre[2]

Bridgeport Instruments, LLC[1] and Pacific Northwest National Laboratory[2]

Sponsored by Army Space and Missile Defense Command

Contract No. W9113M-07-C-0180
Proposal No. BAA07-48

ABSTRACT

Detecting radioactive xenon isotopes in air is one of the methods employed in nuclear explosion monitoring. Using scintillator detectors, the xenon signal is distinguished from background via coincidence spectroscopy. The resulting detectors are complex instruments, with up to 12 scintillators, requiring specialized data acquisition (DAQ) electronics.

Bridgeport Instruments, in collaboration with Pacific Northwest National Laboratory is developing a compact, low-cost DAQ solution. The DAQ module produces raw data, such as energies, arrival times, triggers, etc in real time using digital signal processing embedded in FPGAs. These raw data are available immediately and can be combined in real time by an event builder to form complex events and to check for trigger patterns.

It is a unique characteristic of the DAQ module, that the event builder can be implemented by the customer in an open-source FPGA. In a similar manner, multiple DAQ modules can be connected to a backplane that contains programmable units, such as a trigger FPGA and a communication and control FPGA. All programmable logic on the backplane will be completely open-source.

With this approach we are creating a very versatile platform that uses proprietary code only to generate raw data from the scintillator detector pulses (energies, triggers, etc.). All other aspects, especially those that create a unique instrument from the combination of scintillator detectors are completely under customer control.

In this paper we present the architecture of this new nuclear instrumentation platform.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
SEP 2008		**00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Programmable Front-End Data Acquisition Module for Radio-Xenon Monitors	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Pacific Northwest National Laboratory,PO Box 999,Richland,WA,99352**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Proceedings of the 30th Monitoring Research Review: Ground-Based Nuclear Explosion Monitoring Technologies, 23-25 Sep 2008, Portsmouth, VA sponsored by the National Nuclear Security Administration (NNSA) and the Air Force Research Laboratory (AFRL)

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**8**	

OBJECTIVES

We aim to achieve two distinct goals: to create a commercially viable, general-purpose nuclear instrumentation platform and to create an implementation that will specifically serve the requirements of nuclear explosion monitoring via radio-xenon detection.

In particular we seek to create a platform that is partially open-source, allowing researchers and application engineers to add their own programming and thus create a unique instrument. The front-end modules will directly connect to scintillator detectors and compute raw data in real time. Such raw data include radiation-pulse energies, arrival times, trigger information, and even pulse-shape data. Raw data can be combined on the fly in an adjacent programmable unit, where more complex decisions and data formatting may take place, such as energy-sum triggers, trigger-pattern evaluation, and event building.

A number of front-end modules can be connected to a backplane, which will have completely open-source programming. The backplane is used to control a number of front-end modules and to combine and reformat data, before sending them to the host computer.

To interface with existing data acquisition computers, the front-end modules as well as the back plane will be able to communicate at least via USB and Ethernet.

Our second goal is to use this platform to implement new front-end data acquisition electronics for the Automated Radioxenon Sample/Analyzer (ARSA) radio-xenon detectors. That work will be discussed in a separate paper.

RESEARCH ACCOMPLISHED

The platform concept

The qMorpho has been designed to be an instrument on a PC board. The signal-processing (SP) FPGAs produce raw data, such as energies, etc., for each channel and an application engineer can program the open-source Communication and Control (C&C) FPGA to combine these data at will. This way, the application engineer can create a unique piece of instrumentation without having to divulge any proprietary or classified information to the hardware manufacturer.

The qMorpho data acquisition card can be thought of as a two-tiered system. The first tier is formed by two signal-processing (SP) FPGAs that absorb the signals from a total of four detectors. They operate by examining the digitized waveform of the detector signals and can therefore produce raw data almost as fast as theoretically possible. For each recognized pulse they produce a set of raw data that includes: 1) a trigger pulse indicating the arrival of a signal from the detector; 2) a trigger pulse indicating that the event has been accepted (i.e. it is not piled up or out of range); 3) a measure of its energy; 4) a measure of its arrival time; 5) a pile up measure; and 6) a pulse shape measure.

In standard configuration, the SP FPGAs will use these raw data to: 1) send trigger information to the C&C FPGA; 2) make energy, time and pulse-shape data available to the C&C FPGA; 3) accumulate statistics information in four counters per channel, measuring data acquisition time, accepted-event rates, input pulse rates, and trigger dead time; 4) accumulate one energy histogram for each channel; 5) accumulate list mode data; and 6) store individual event waveforms (oscilloscope function). In other words, in standard configuration the qMorpho acts like four independent DAQ channels providing triggering, energy histogramming, list mode, trace capture and even on-the-fly pulse shape analysis for particle discrimination, such as n/γ discrimination in liquid scintillators.

However, the true power of a multichannel DAQ system rests in its ability to combine the raw data event-by-event and in real time. Of course, users can only harness this power if they have programmatic access to the raw data. This is the key to creating a novel and unique radiation detection instrument.

In the past, scientists and engineers would use dedicated nuclear incident monitor (NIM) modules to create the raw data and use cabling and logic modules to realize a particular experiment. Now we can do better—in a much smaller space and at a much lower power consumption.

The C&C FPGA, which has access to all raw data, is open to custom programming. Almost arbitrarily complex trigger logic can be implemented with a few lines of verilog code. Secondary data sets, such as sum-energy histograms, or list mode data depending on coincidence time windows can be generated here. Data-formatting, zero suppression, and even event-building can be implemented here.

Just like its predecessor, the single-channel eMorpho, the four-channel qMorpho is supplied with an application programmer's interface (API) that is written in plain C with a minimal dependence on the operating system. The one notable exception is the single function that actually moves data across the physical interface. This design has made it possible to implement the eMorpho API on diverse platforms including Windows, Unix, Linux, QNX, as well as PIC and Rabbit 8-bit and 16-bit microprocessors using USB, UART/Ethernet or SPI physical interfaces.

qMorpho Summary Description

The first front-end module of this new nuclear instrumentation platform is the **qMorpho**. Based on the successful single-channel eMorpho, this unit is designed for operation with four independent scintillator detectors. For each detector there are two functional groups. One accepts the photomultiplier tube (PMT) signal directly from the PMT anode, and sends it through a switchable gain stage directly to a waveform digitizing ADC, without any pulse shaping. A signal-processing FPGA picks up the data stream from the ADC. Using digital signal processing, it analyses the digital image of the analog input pulse and computes the raw data such as energies, triggers, time stamps, pulse shapes, etc.

The second functional group consists of support elements, such as a 12-bit digital to analog converter (DAC) and a digital interface to control plug-on PMT HV-supplies (TwinBase), and a digital interface to the temperature sensor of the TwinBase. A set of general purpose I/O pins (GPIO) enhance the ability of the front-end section to interface with more sophisticated detector monitors and controls, e.g., for HV and power consumption. Finally, a set of trigger inputs and outputs facilitate integration and cooperation with secondary detectors.

There are three programmable units (all XC3S-400 FPGAs) on a qMorpho. Two of these equally share the task of computing the raw data for the four detectors.

qMorpho analog sections

The qMorpho can process data from four scintillator detectors simultaneously. For each detector, the analog signal is accepted directly from the PMT anode. The first amplifier is a current-to-voltage converter with a computer-controlled switchable gain. The available gain, expressed as a transimpedance (V/A = Ω), ranges from 100 Ω to 10,100 Ω. A differential-output OP-Amp presents this signal to a waveform digitizing ADC with an input voltage range of close to 1.0V. Hence, a full scale signal corresponds to a 10-mA peak anode current at the lowest gain and 0.10 mA at the highest gain. This range of maximum anode pulse currents is well matched to the characteristics and capabilities of the vast majority of photomultipliers.

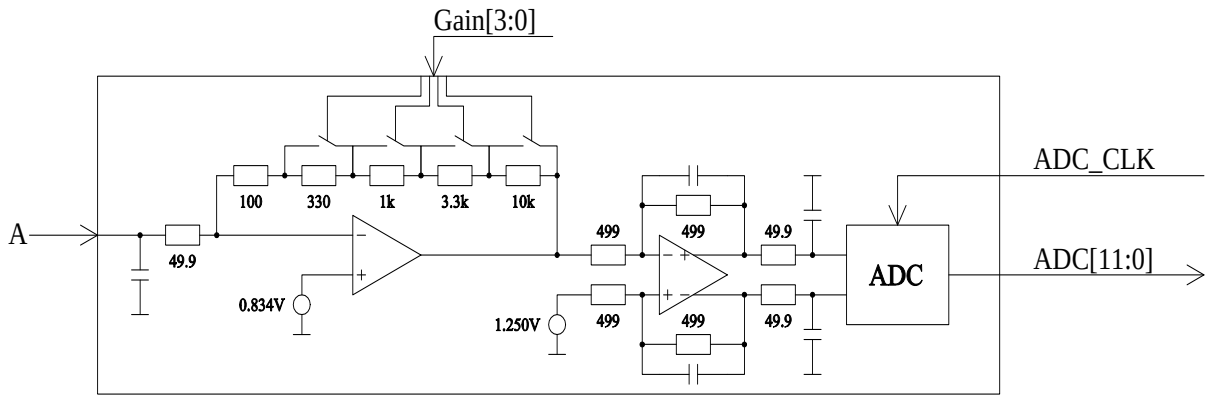

Figure 1. qMorpho analog section, 1 of 4 channels. The qMorpho receives the PMT anode signal and converts it into a differential signal which is fed directly to the waveform digitizing ADC without integration or other pulse shaping. There are four identical analog channels.

qMorpho architecture

Figure 2 shows the architecture of the qMorpho DAQ card. Functionally, there are four independent acquisition channels, each serving one scintillator detector. The proprietary component of the digital signal processing is implemented in the two signal-processing FPGAs (SP0, SP1). The two connect to an open-source C&C FPGA, which can be accessed by a host computer through a number of different physical communication channels. At the minimum, these include a USB-1.1 interface, a UART/Ethernet interface as well as a connection to a 16-bit wide backplane data bus.

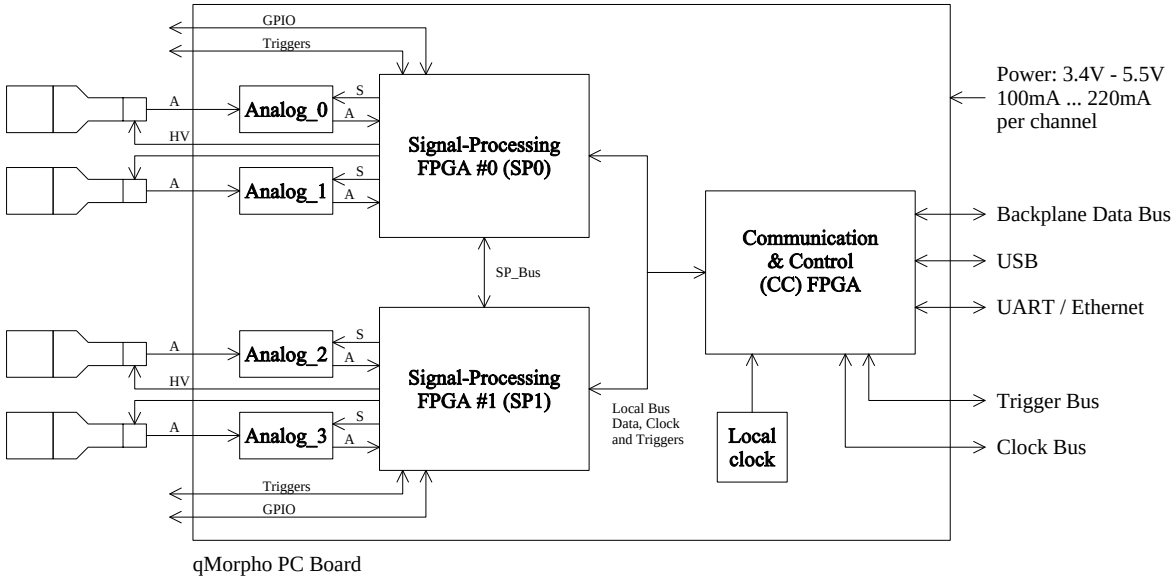

Figure 2. qMorpho architecture.

The signal-processing FPGAs generate triggers for each recognized scintillator pulse and send these to the C&C FPGA. The C&C FPGA can be used to validate events according to user-programmed trigger patterns and return validation triggers to the SP FPGAs. In a larger system, the C&C FPGA may pass the trigger information on to a backplane, where a dedicated trigger FPGA collects information from more than one qMorpho, makes its decision and returns validation triggers. The C&C FPGAs would then route the validation triggers down to the SP FPGAs.

The qMorpho has an onboard 24 MHz clock oscillator. A digital clock manager (DCM) in the C&C FPGA transforms the local clock into a 10_MSPS to 100_MSPS ADC sampling clock. Every operation on the card, with the exception of the USB micro-controller, is synchronous with the ADC clock. The C&C sends the ADC clock down to the SP FPGAs who supply the actual clock signals to the waveform digitizing ADCs.

Alternatively, the clock can be generated by a backplane and received by the qMorpho. The qMorpho can even act as a clock repeater, restoring clock amplitudes and duty cycles.

The qMorpho accepts power through a single input. The highest regulated voltage on the qMorpho board is 3.30 V. Using low-dropout regulators the qMorpho can accept input voltages from as low as 3.4 V up to 5.5 V. Power consumption can be below a 100 mA/channel when using 10-bit ADCs and rises to a maximum of 220 mA/channel when using 12-bit ADCS at 80 MSPS.

qMorpho communication

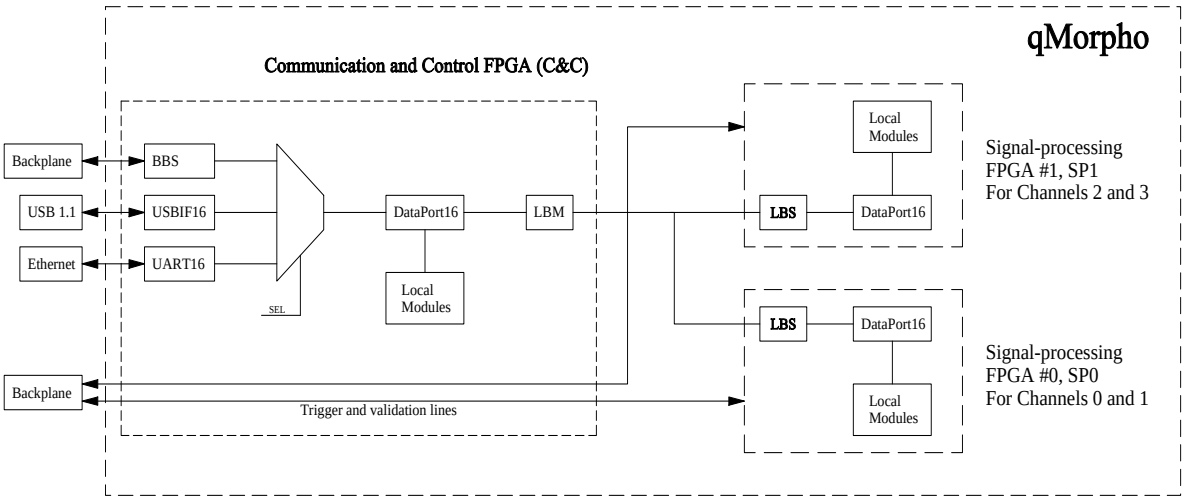

Figure 3. Summary drawing of the qMorpho communication data paths. The C&C FPGA has access to all raw data generated by the signal processing FPGAs. The C&C FPGA is open-source and can be reprogrammed by application engineers to combine raw data as required, for instance to enforce complex trigger patterns, to build sum spectra, or in other ways create a unique instrument or

All FPGAs on the qMorpho (and the associated backplane) use the same communication method. The central unit is the DataPort16. It receives 16-bit wide data and a set of control signals from an upstream interface module. For instance, figure 3 shows three interface modules able to communicate with the DataPort16 of the C&C FPGA: a backplane bus server, a USB interface and a UART/Ethernet interface. The C&C DataPort16 can direct the data flow to local data sinks and sources (timers, counters, version registers, etc.) or direct the data flow through a local bus master (LBM) onto the local data bus running between the three on board FPGAs. On the other end the signal-processing FPGAs connect to the local bus via a local bus server (LBS) that feeds into their DataPort16. The SP DataPort16 only connects to local modules which hold control/action registers and results from signal processing activities as shown in figure 4.

All communications start with writing data to the target FPGA. Every writing sequence begins with sending four 16-bit packet header words (PH) followed by a varying number of data words; i.e. the payload. The packet header information is used for routing the data to the intended recipient and to control the behavior of the target FPGA as well as the route-through devices. Table 1 illuminates the protocol being used.

	15							**8**	**7**							**0**
PH_0	FS			R	WORDS				SN							
PH_1	CM				MA				PAGE						A	D
PH_2																
PH_3																

Table 1: Content of the four packet headers prepended to each data packet.

Standard code makes use of only the first two packet headers, while the others are reserved for future use.
The 2-bit-wide field FS is used to select the target FPGA. SN is the slot number in a backplane-based system. CM is

the channel mask of addressed channels on the qMorpho. MA is the module address in whichever FPGA has been addressed. A and D are bits governing clearing the local address counter used for addressing data in the FPGA's modules. PAGE is the starting page (256-byte page size) for the next read operation. When using the UART/ Ethernet connection, WORDS determines how many words the unit will send back after receiving a read request. Setting the return bit (R) forces the unit to start sending data at the end of the write cycle. The packet header is followed by 28 data words.

Writing to a qMorpho

Writing data to the qMorpho is straightforward. Four packet header words are loaded with the necessary routing information and transmitted first. They are then followed by 28 data words. Most often, the write is directed to one or more control register sets to program various aspects of the digital signal processing as well as the PMT high voltage and a built-in electronic pulser. The host is always writing 32 words in one try. When writing to the action registers (that control DAQ start and memory clearing) only the first four words of the payload are stored, and the rest are discarded.

Consider writing data via a backplane to a qMorpho. The C&C FPGA on the backplane will recognize from the FS mask that it is not the intended recipient of the data and will send them to the qMorpho residing in the slot number SN. A broadcast to all slots is achieved by using the special slot number 255.

The C&C FPGA on the addressed qMorpho can tell from the content of the FS field if it is the intended recipient or if data should be sent on to one of the SP FPGAs. If the data are meant for the C&C FPGA, they will be stored in a local module (mostly a set of registers). The module address is encoded in the 4-bit-wide field MA, which allows for 16 different modules.

If the data are meant for one or more channels in the SP FPGA, the qMorpho C&C FPGA uses the CM field to determine the target channels and accordingly sends the data to the intended SP FPGA(s). It is possible to send data to more than one processing channel at the same time by setting more than one bit in the CM field.

Reading from a qMorpho

The method of reading from a qMorpho depends somewhat on the communication method used. For methods that provide a physical set of lines distinguishing reads from writes (R/W# line, or RX and TX lines) we first write to the qMorpho to select the target FPGA, target module, and the starting memory address. Then we issue the read command and read as many bytes as necessary. When operating through the UART (or an SPI) interface where there is only a Data_In and a Data_Out line, we write to the qMorpho as before but add the read (R) bit and the number of words to be read (WORDS) to the packet header as shown in Table 1. In this case the qMorpho will start sending data through the serial Data_Out automatically after the write cycle has been completed.

qMorpho Communication Interface

The qMorpho is set up to accept communication through three different physical interfaces: The backplane bus, a USB 1.1 interface, and a UART. The first is used when the qMorpho is part of a larger system and is connected to a backplane or an equivalent system controller. Alternatively, the backplane bus interface can serves as a convenient interface to an embedded local microcontroller or single board computer. The second is a practical option for individual qMorpho cards controlled by a local host computer in the laboratory or in backpack-sized portable systems. The UART interface is used in conjunction with Ethernet-to-UART bridges where the Ethernet connection can be established through wired or wireless Ethernet modules at the user's discretion. Only the USB interface requires operating system dependent and proprietary driver software—the manufacturer's class driver. All other interfaces are open-source and can be manipulated as the application engineers see fit.

The Signal-Processing FPGAs

The signal-processing FPGA provides identical real-time digital signal processing for all four channels. Implemented on a single silicon chip they deliver the functionality of many NIM modules. Figure 4 shows the

minimum setup as provided by the factory. Standard data acquisition capabilities include triggering, energy measurement, histogramming, oscilloscope function, list mode data and more.

At the same time, the qMorpho can power plug-on PMT HV-supplies (TwinBase) and the host can control the high voltage through registers in the SP FPGA. The host can also read the temperature of the TwinBase through the SP FPGA. Finally, there is a programmable pulser module for single and double pulses that can be used to drive an LED for optical gain measurements and gain stabilization.

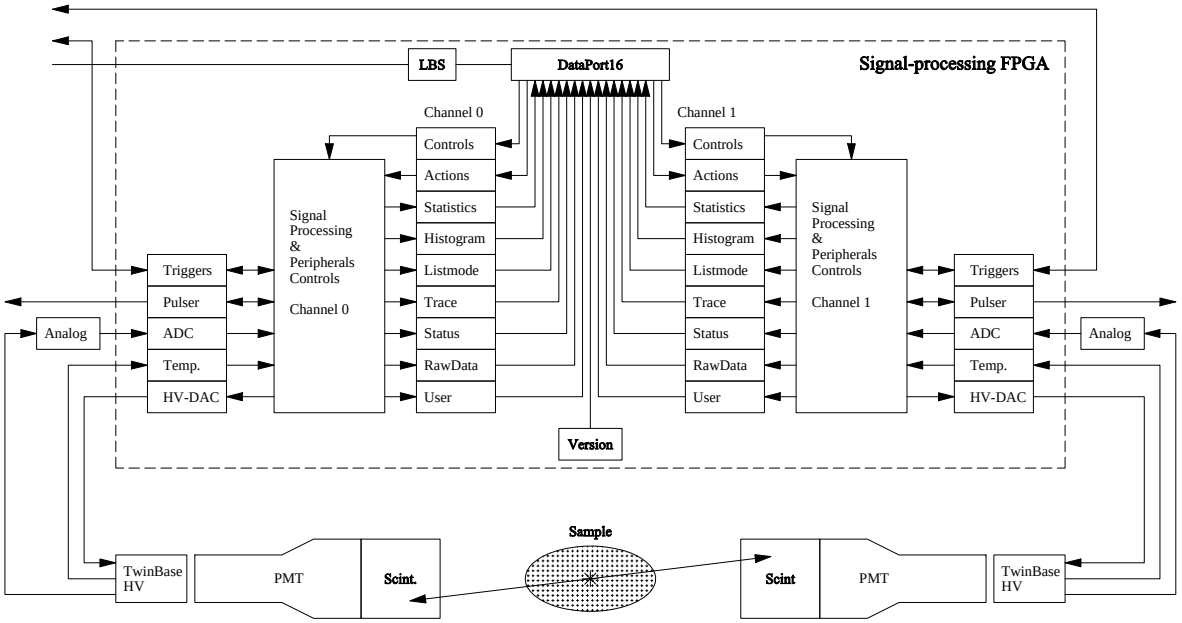

Figure 4. Summary drawing of the qMorpho signal processing FPGAs.

Factory-installed modules in the SP FPGAs will at least include the ones shown in Figure 4. The control register (28 words per channel) contain all the parameters required for the digital signal processing of the detector signals as well as the high voltage setting for the TwinBase plug on PMT high-voltage supplies. The action registers (4 words) contain bit-fields governing the data acquisition. These include one-time action, self-clearing bits for clearing histogram memory, resetting statistics counters and the like. The second action register carries continued-action bits such as those used to start and stop all kinds of data acquisition, for instance histogramming, oscilloscope function, and list-mode data acquisition. A user module provides memory for 1,024 data words provided by the host application. These three modules are write-only; all others are read only.

The statistics module consists of a set of counters collecting all the necessary information to accurately measure incoming pulse rates and the rate of accepted events. The histogramming module provides 8K x 32-bit histogramming memory for each channel. The list mode module can store energies and time stamps for 340 events before its buffer is full. The Trace module can record 1,024 waveform samples on a number of trigger conditions. It can shift the recorded waveform in memory to provide access to the pretrigger baseline. The status module reports the data acquisition status for the different DAQ types in a 16-bit status register. The RawData module reports event data (energies and time stamps) at the lower addresses followed by slower changing data such as DC-baselines and TwinBase temperatures. A host process (on the host computer, the backplane or in the qMorpho C&C FPGA) can choose to read all or just a subset of the RawData module.

For all registers and modules mentioned so far, there is one copy for each DAQ channel. The version module is the only one for which there is only one per SP FPGA. It reports information about the clock speed and firmware version.

CONCLUSIONS AND RECOMMENDATIONS

In this paper we have described and documented our approach to providing a data acquisition solution that is general enough to become a commercial off-the-shelf (COTS) product, yet flexible enough to satisfy the particular requirements of the ARSA radioxenon stations. Our approach uses a judicious combination of proprietary and open-source firmware to allow scientists and application engineers to customize the instrument performance without having to share proprietary or classified information with the manufacturer. Its open-source control and communication FPGA allows application engineers to combine raw data, such as energies, triggers and time stamps, at will and in real time on an event-by-event basis. An operating-system independent application programmer's interface ensures that the qMorpho is compatible with almost every computer and microcontroller on the market. The qMorpho design is open-source where it matters, giving application engineers full control over how to combine the raw data from four independent nuclear radiation detectors, in real time and on an event-by-event basis. When operating with a backplane, application engineers can customize the operation and performance of larger detector arrays without having to rely on the manufacturer to provide custom programming.

REFERENCES

Bowyer, T.W., K. H. Abel, C. W. Hubbard, A. D. McKinnon, M. E. Panisko, R. W. Perkins, P. L. Reeder, R. C. Thompson, and R.A. Warner (1998). “Automated separation and measurement of radioxenon for the Comprehensive Test Ban Treaty, *Journal of Radioanalytical and Nuclear Chemistry*, 235: 1-2: 77–81.